P☉RTAL

USPTO

Search: ◉ The ACM Digital Library   ○ The Guide

java reflection bytecode xml   SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: **java reflection bytecode xml**   Found **10,264** of **211,032**

Sort results by   relevance   ▼   ❖ Save results to a Binder

Display results   expanded form   ▼   ❓ Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200   Result page: **1**   2   3   4   5   6   7   8   9   10   next
Best 200 shown   Relevance scale ▢▢▨▨■

1 JTL: the Java tools language   ■

Tal Cohen, Joseph (Yossi) Gil, Itay Maman

October 2006 **ACM SIGPLAN Notices , Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**, Volume 41 Issue 10

Publisher: ACM Press

Full text available: 📄 pdf(386.63 KB)   Additional Information: full citation, abstract, references, citings, index terms

We present an overview of JTL (the Java Tools Language, pronounced "Gee-tel"), a novel language for querying JAVA [8] programs. JTL was designed to serve the development of source code software tools for JAVA, and as a small language which to aid programming language extensions to JAVA. Applications include definition of pointcuts for aspect-oriented programming, fixing type constraints for generic programming, specification of encapsulation policies, definition of micro-patterns, etc. We argue ...

Keywords: declarative programming, reverse engineering

2 Performance of service oriented systems: Speed-up SOAP processing by data mapping template   ■

Wei Jun, Hua Lei, Niu Chunlei

May 2006 **Proceedings of the 2006 international workshop on Service-oriented software engineering SOSE '06**

Publisher: ACM Press

Full text available: 📄 pdf(245.04 KB)   Additional Information: full citation, abstract, references, index terms

Web Services is gaining popularity in distributed computing due to its loosely-coupled, high-interoperable and platform-independent characteristics. However, web services suffers performance penalty because XML based SOAP is used to specify wire message format, and SOAP processing largely affects the performance of web services. In this paper, we identify that data model mapping between XML data and Java data is the main impact factor on performance, and propose a new paradigm of data model mapp ...

Keywords: SOAP, context free grammar, data mapping template, dynamic early binding, web services

3

Join point selectors

Cristiano Breuel, Francisco Reverbel

March 2007 **Proceedings of the 5th workshop on Engineering properties of languages and aspect technologies SPLAT '07**

Publisher: ACM Press

Full text available: 📄 pdf(187.71 KB)   Additional Information: full citation, abstract, references, index terms

One of the main issues in modern aspect-oriented programming languages and frameworks is the expressiveness of the pointcut language or mechanism. The expressiveness of pointcut languages directly impacts pointcut quality, a property that can be decisive for the effectiveness of aspect implementations. In this paper we propose join point selectors as a simple extension mechanism for enriching current pointcut languages with constructs that play the role of "new primitive pointcuts". Join poin ...

Keywords: AOP, aspect-oriented programming, extensibility, join point selectors, pointcut languages, semantic pointcuts

4   When Role Models Have Flaws: Static Validation of Enterprise Security Policies

Marco Pistoia, Stephen J. Fink, Robert J. Flynn, Eran Yahav

May 2007 **Proceedings of the 29th International Conference on Software Engineering ICSE '07**

Publisher: IEEE Computer Society

Full text available: 📄 pdf(255.02 KB)   Additional Information: full citation, abstract, index terms

Modern multiuser software systems have adopted Role- Based Access Control (RBAC) for authorization management. This paper presents a formal model for RBAC policy validation and a static-analysis model for RBAC systems that can be used to (i) identify the roles required by users to execute an enterprise application, (ii) detect potential inconsistencies caused by principal-delegation policies, which are used to override a user's role assignment, (iii) report if the roles assigned to a user by a g ...

5   Service security: A concrete solution for web services adaptability using policies and aspects

Fabien Baligand, Valérie Monfort

November 2004 **Proceedings of the 2nd international conference on Service oriented computing ICSOC '04**

Publisher: ACM Press

Full text available: 📄 pdf(368.60 KB)   Additional Information: full citation, abstract, references, citings, index terms, review

Traditional middleware is usually developed on monolithic and non-evolving entities, resulting in a lack of flexibility and interoperability. Among current architectures, Service Oriented Architectures aim to easily develop more adaptable Information Systems. Most often, Web Service is the fitted technical solution which provides the required loose coupling to achieve such architectures. However there is still much to be done in order to obtain a genuinely flawless Web Service, and current ma ...

Keywords: adaptability, aspect oriented programming, reusability, service, service oriented architecture, web service

6   Fast online pointer analysis

Martin Hirzel, Daniel Von Dincklage, Amer Diwan, Michael Hind

April 2007 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 29 Issue 2

Publisher: ACM Press

Full text available: pdf(430.96 KB)   Additional Information: full citation, abstract, references, index terms

Pointer analysis benefits many useful clients, such as compiler optimizations and bug finding tools. Unfortunately, common programming language features such as dynamic loading, reflection, and foreign language interfaces, make pointer analysis difficult. This article describes how to deal with these features by performing pointer analysis online during program execution. For example, dynamic loading may load code that is not available for analysis before the program starts. Only an online an ...

Keywords: Pointer analysis, class loading, native interface, reflection

7 Partial behavioral reflection: spatial and temporal selection of reification

Éric Tanter, Jacques Noyé, Denis Caromel, Pierre Cointe
October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11
Publisher: ACM Press

Full text available: pdf(261.44 KB)   Additional Information: full citation, abstract, references, citings, index terms

Behavioral reflection is a powerful approach for adapting the behavior of running applications. In this paper we present and motivate *partial behavioral reflection*, an approach to more efficient and flexible behavioral reflection. We expose the *spatial* and *temporal* dimensions of such reflection, and propose a model of partial behavioral reflection based on the notion of *hooksets*. In the context of Java, we describe a reflective architecture offering appropriate interf ...

Keywords: aspect-oriented programming, open systems, reflection

8 Bytecode fetch optimization for a Java interpreter

Kazunori Ogata, Hideaki Komatsu, Toshio Nakatani
October 2002 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 36 , 37 , 30 Issue 5 , 10 , 5
Publisher: ACM Press

Full text available: pdf(1.16 MB)   Additional Information: full citation, abstract, references, citings, index terms

Interpreters play an important role in many languages, and their performance is critical particularly for the popular language Java. The performance of the interpreter is important even for high-performance virtual machines that employ just-in-time compiler technology, because there are advantages in delaying the start of compilation and in reducing the number of the target methods to be compiled. Many techniques have been proposed to improve the performance of various interpreters, but none of ...

Keywords: Java, PowerPC, bytecode interpreter, performance, pipelined interpreter, stack caching, superscalar processor

9 Dynamic metrics for java

Bruno Dufour, Karel Driesen, Laurie Hendren, Clark Verbrugge
October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11
Publisher: ACM Press

Full text available: pdf(222.67 KB)   Additional Information: full citation, abstract, references, citings, index terms

In order to perform meaningful experiments in optimizing compilation and run-time system design, researchers usually rely on a suite of benchmark programs of interest to the optimization technique under consideration. Programs are described as *numeric*, *memory-intensive*, *concurrent*, or *object-oriented*, based on a qualitative appraisal, in some cases with little justification. We believe it is beneficial to quantify the behaviour of programs with a concise and precisely ...

Keywords: Java, dynamic metrics, execution traces, optimization, profiling, program analysis, software metrics

10 Practical extraction techniques for Java

Frank Tip, Peter F. Sweeney, Chris Laffra, Aldo Eisma, David Streeter
November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 6
Publisher: ACM Press

Full text available: pdf(1.01 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

Reducing application size is important for software that is distributed via the internet, in order to keep download times manageable, and in the domain of embedded systems, where applications are often stored in (Read-Only or Flash) memory. This paper explores extraction techniques such as the removal of unreachable methods and redundant fields, inlining of method calls, and transformation of the class hierarchy for reducing application size. We implemented a number of extraction techniques in < ...

Keywords: Application extraction, call graph construction, class hierarchy transformation, packaging, whole-program analysis

11 Practical experience with an application extractor for Java

Frank Tip, Chris Laffra, Peter F. Sweeney, David Streeter
October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10
Publisher: ACM Press

Full text available: pdf(2.31 MB)   Additional Information: full citation, abstract, references, citings, index terms

Java programs are routinely transmitted over low-bandwidth network connections as compressed class file archives (i.e., zip files and jar files). Since archive size is directly proportional to download time, it is desirable for applications to be as small as possible. This paper is concerned with the use of program transformations such as removal of dead methods and fields, inlining of method calls, and simplification of the class hierarchy for reducing application size. Such "extract ...

12 The DaCapo benchmarks: java benchmarking development and analysis

Stephen M. Blackburn, Robin Garner, Chris Hoffmann, Asjad M. Khang, Kathryn S. McKinley, Rotem Bentzur, Amer Diwan, Daniel Feinberg, Daniel Frampton, Samuel Z. Guyer, Martin Hirzel, Antony Hosking, Maria Jump, Han Lee, J. Eliot, B. Moss, Aashish Phansalkar, Darko Stefanović, Thomas VanDrunen, Daniel von Dincklage, Ben Wiedermann
October 2006 **ACM SIGPLAN Notices , Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**, Volume 41 Issue 10
Publisher: ACM Press

Full text available: pdf(3.11 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

Since benchmarks drive computer science research and industry product development, which ones we use and how we evaluate them are key questions for the community. Despite complex runtime tradeoffs due to dynamic compilation and garbage collection required for Java programs, many evaluations still use methodologies developed for C, C++, and Fortran. SPEC, the dominant purveyor of benchmarks, compounded this problem by institutionalizing these methodologies for their Java benchmark suite. This pap ...

Keywords: DaCapo, Java, SPEC, benchmark, methodology

13 Programming techniques: Manipulation of Java agent bytecode to add roles

Giacomo Cabri, Luca Ferrari, Letizia Leonardi

June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java PPPJ '03**

Publisher: Computer Science Press, Inc.

Full text available: pdf(215.46 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>

Roles are a powerful paradigm to develop distributed applications based on agents, especially when they are in need of interacting with other entities. An agent-oriented approach requires that roles are conceived as first-class entities, and at the same time that roles are dynamically embedded into agents at runtime. In this paper we propose an approach that addresses such requirements, enabling Java agents to dynamically assume roles. We present a mechanism that modifies the agent bytecode to a ...

14 Compilation: Reflective program generation with patterns

Manuel Fähndrich, Michael Carbin, James R. Larus

October 2006 **Proceedings of the 5th international conference on Generative programming and component engineering GPCE '06**

Publisher: ACM Press

Full text available: pdf(200.25 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

Runtime reflection facilities, as present in Java and .NET, are powerful mechanisms for inspecting existing code and metadata, as well as generating new code and metadata on the fly. Such power does come at a high price though. The runtime reflection support in Java and .NET imposes a cost on all programs, whether they use reflection or not, simply by the necessity of keeping all metadata around and the inability to optimize code because of future possible code changes. A second---often overlook ...

Keywords: generative programming, patterns, reflection, templates

15 Workshop on Dynamic Analysis (WODA): Dynamic analysis of java applications for multithreaded antipatterns

S. Boroday, A. Petrenko, J. Singh, H. Hallal

May 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the third international workshop on Dynamic analysis WODA '05**, Volume 30 Issue 4

Publisher: ACM Press

Full text available: pdf(92.55 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

Formal verification is not always applicable to large industrial software systems due to scalability issues and difficulties in formal model and requirements specification. The scalability and model derivation problems could be alleviated by runtime trace analysis, which combines both testing and formal verification. We implement and compare an ad-hoc custom approach and a formal approach to detect common bug patterns in

multithreaded Java software. We use the tracing platform of the Eclipse IDE ...

Keywords: Java, antipatterns, bug patterns, bytecode, instrumentation, multithreading

16 Extending Java for high-level Web service construction

Aske Simon Christensen, Anders Møller, Michael I. Schwartzbach

November 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 6

Publisher: ACM Press

Full text available: pdf(947.02 KB)   Additional Information: full citation, abstract, references, citings, index terms

We incorporate innovations from the <bigwig> project into the Java language to provide high-level features for Web service programming. The resulting language, JWIG, contains an advanced session model and a flexible mechanism for dynamic construction of XML documents, in particular XHTML. To support program development we provide a suite of program analyses that at compile time verify for a given program that no runtime errors can occur while building documents or receiving form input, and ...

Keywords: Interactive Web services, XML, data-flow analysis

17 Ibis: an efficient Java-based grid programming environment

Rob V. van Nieuwpoort, Jason Maassen, Rutger Hofman, Thilo Kielmann, Henri E. Bal

November 2002 **Proceedings of the 2002 joint ACM-ISCOPE conference on Java Grande JGI '02**

Publisher: ACM Press

Full text available: pdf(120.05 KB)   Additional Information: full citation, abstract, references, citings, index terms, review

In computational grids, performance-hungry applications need to simultaneously tap the computational power of multiple, dynamically available sites. The crux of designing grid programming environments stems exactly from the dynamic availability of compute cycles: grid programming environments (a) need to be *portable* to run on as many sites as possible, (b) they need to be *flexible* to cope with different network protocols and dynamically changing groups of compute nodes, while (c) t ...

Keywords: Java, grid computing, performance, portability

18 Programming for separation of concerns (PSC): Policy-driven reflective enforcement of security policies

Ian Welch, Fan Lu

April 2006 **Proceedings of the 2006 ACM symposium on Applied computing SAC '06**

Publisher: ACM Press

Full text available: pdf(105.96 KB)   Additional Information: full citation, abstract, references, index terms

Practical experience has shown that separating security enforcement code from functional code using separation of concerns techniques such as behavioural reflection leads to improvements in code undestandability and maintainability. However, using these techniques at requires providing a consistent and declarative way to specify policies. We have developed a prototype tool that allows the use of Ponder policies that are enforced by the Kava metaobject protocol. This prototype translates high-lev ...

Keywords: reflection, security policies

19 A comprehensive approach for the development of modular software architecture description languages

Eric M. Dashofy, André van der Hoek, Richard N. Taylor

April 2005 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,
Volume 14 Issue 2

Publisher: ACM Press

Full text available: pdf(3.51 MB)   Additional Information: full citation, abstract, references, citings, index terms

Research over the past decade has revealed that modeling software architecture at the level of components and connectors is useful in a growing variety of contexts. This has led to the development of a plethora of notations for representing software architectures, each focusing on different aspects of the systems being modeled. In general, these notations have been developed without regard to reuse or extension. This makes the effort in adapting an existing notation to a new purpose commensurate ...

Keywords: ArchStudio 3, Architecture description languages, XML, xADL 2.0

20 Aspect-oriented programming with adaptive methods

Karl Lieberherr, Doug Orleans, Johan Ovlinger

October 2001 **Communications of the ACM**, Volume 44 Issue 10

Publisher: ACM Press

Full text available: pdf(111.04 KB)   Additional Information: full citation, appendices and supplements,
html(13.21 KB)   references, cited by, index terms

Results 1 - 20 of 200   Result page: **1**   2   3   4   5   6   7   8   9   10   next

Google

java reflection bytecode persistence class | Search | Advanced Search
                                                             Preferences
                                               New! View and manage your web history

Web   Results **1 - 10** of about **292,000** for **java reflection** bytecode **persistence class**. (**0.18** seconds)

com.sleepycat.persist (Oracle - Berkeley DB **Java** Edition API)
Bytecode Enhancement. The **persistent** fields of a **class** may be private, ... In applications
that are CPU bound, avoiding **Java reflection** can have a ...
www.oracle.com/technology/documentation/berkeley-
db/je/**java**/com/sleepycat/persist/package-summary.html - 35k - Cached - Similar pages

Chapter 1. Introduction
The only real requirement is that each **persistent class** have a default constructor. ...
provides fully optimized bindings that do not use **Java reflection**. ...
www.oracle.com/technology/documentation/berkeley-
db/je/**Persistence**API/introduction.html - 11k - Cached - Similar pages
[More results from www.oracle.com]

Java Persistent Objects - **Bytecode** Enhancement
This means that you get transparent **persistence**, and your **classes** always remain your
classes. ORM tools that use a mix of **reflection** and/or proxies are not ...
www.jpox.org/docs/1_2/**bytecode**_enhancement.html - 17k - Cached - Similar pages

The **Java Persistence** Landscape
Then, for **persistent classes**, you'd run that **class** file through a JDO **byte** ... Hibernate
uses **Java reflection** instead of **byte code** enhancement to achieve ...
dev2dev.bea.com/pub/a/2003/07/**Java_Persistence**.html - 43k - Cached - Similar pages

Javassist: **Java Bytecode** Engineering Made Simple @ **JAVA** ...
The first part of the Javassist API is similar to the **Java Reflection** API. ... Javassist is not
the first **class** library for writing a **bytecode** translator. ...
java.sys-con.com/read/38672.htm - 78k - Cached - Similar pages

JEE and Oracle Fusion Middleware Weblog: JavaOne 2007 event ...
main object is the **persistent class**; the domain object is persisted into the DB ... JPOX
uses **byte code**; Toplink essentials uses **byte code**, **reflection** for ...
emarcoux.blogspot.com/2007/05/**java**one-2007-event-architecture-of.html - 18k -
Cached - Similar pages

[PDF] Architecture of **Java Persistence** Implementations
File Format: PDF/Adobe Acrobat - View as HTML
Fields behave according to **Java** platform access. modifiers. >Private, protected. >
Persistence provider uses **reflection** or **byte code** ...
www.sda-india.com/conferences/jax-india/sessions/Craig_Russell/jax2007architecture.pdf -
Similar pages

Javassist: **Java bytecode** engineering made simple: for source-level ...
The first part of the Javassist API is similar to the **Java Reflection** API. It allows you to view
the structure of a **Java class** before it is loaded by the ...
goliath.ecnext.com/coms2/gi_0199-674905/**Java**ssist-**Java**-**bytecode**-engineering-
made.html - 24k - Cached - Similar pages

JDBCPersistence - Object Relational Mapping Framework
As opposed to using **reflection**, the approach of generating **bytecode** is apt to ... Similar to
generating full implementation for **Java** interface **class** that ...

www.jdbc**persistence**.org/ - 11k - Cached - Similar pages

[PDF] Exploiting **Reflection** to Add **Persistence** and Query Optimization to ...
File Format: PDF/Adobe Acrobat - View as HTML
bytecodes. We are extending **Java** with true linguistic **reflection**, that is, the ability to
change the definition of a. **class** and its methods at run-time (or ...
www.cs.umass.edu/~moss/papers/pos-1998.pdf - Similar pages

1 2 3 4 5 6 7 8 9 10   **Next**

Download Google Pack: free essential software for your PC

---

java reflection bytecode persistence   Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

---

©2007 Google - Google Home - Advertising Programs - Business Solutions - About Google